

Peeks, Pokes and Pointers

Apple® Zero-Page

DECIMAL	HEX
32 Text Window Left-Edge (0-39)	\$20
Example: POKE 32, 5 freezes the left 5 columns of text. Note: Do not use without also adjusting text WIDTH (location 33) to 40 minus the left edge value (POKE 33, 35 for this example).	
33 Text Window Width (1-40)	\$21
34 Text Window Top-Edge (0-23)	\$22
35 Text Window Bottom (1-24)	\$23
36 Horizontal Cursor-Position (0-39)	\$24
Examples: If PEEK(36) =8, then the cursor is in the 9th column. POKE 36, 19 is the same as HTAB 20 . Useful with 80-column printers or screens; for printing text beyond the 40-character range of HTAB (example: POKE 36, 70 for column-71).	
37 Vertical Cursor-Position (0-23)	\$25
Examples: If PEEK(37) =5, then the cursor is on the 6th line. POKE 37, 5 has the same effect as VTAB 6 .	
43 Boot Slot * 16 (after boot)	\$2B
44 Lo-Res Line End-Point	\$2C
48 Lo-Res COLOR * 17	\$30
50 Text Output Format	\$32
POKE 50, 63=INVERSE, POKE 50, 255=NORMAL, POKE 50, 127=FLASH (for ASCII 64-95).	
51 Prompt-Character	\$33
Note: POKE 51,0: GOTO line# will prevent a false NOT DIRECT COMMAND sometimes obtained with GOTO line# alone.	
78-79 Random-Number Field	\$4E.4F
103-104 Start of Applesoft Program	\$67.68
Note: FP sets start-of-program to normal 2049. Note: To LOAD a program at a non-standard location LOC— POKE 103, LOC-INT(LOC/256)*256; POKE 104, INT(LOC/256): POKE LOC-1, 0 Then LOAD PROGRAM	
105-106 LOMEM	\$69.6A
Note: LOMEM is the Start of Variable-Space, equivalent to End-of-Program (+1 or +2) unless changed with LOMEM: command.	
107-108 Start of Array-Space	\$6B.6C
109-110 End of Array-Space	\$6D.6E
111-112 Start of String-Storage	\$6F.70
115-116 HIMEM	\$73.74
Note: HIMEM-1 is the highest address available for use by an Applesoft program. May be changed with HIMEM: command.	
117-118 Line# Being Executed	\$75.76
119-120 Line# where Program Stopped ..	\$77.78
121-122 Address of Line Executing	\$79.7A
123-124 Current DATA Line#	\$7B.7C
125-126 Next DATA Address	\$7D.7E
127-128 INPUT or DATA Address	\$7F.80
129-130 Last-Used Variable Name	\$81.82
Example: VAR\$=CHR\$(PEEK(129))+CHR\$(PEEK(130))	
131-132 Last-Used-Variable Address	\$83.84
175-176 End of Applesoft Program	\$AF.B0
Note: Normally corresponds to LOMEM.	
214 RUN Flag	\$D6
Example: POKE 214, 255 makes any command RUN a program.	
216 ONERR Flag	\$D8
Example: POKE 216, 0 cancels the ONERR function.	
218-219 Line# of ONERR Error	\$DA.DB
222 ONERR Error Code	\$DE
PEEK(222) Codes (DOS errors in <i>italic</i>)— 0=?NEXT Without FOR 16=?Syntax Error (FP) 1=Language Not Available 22=?RETURN Without GOSUB 2 or 3=Range Error 42=?Out of DATA 4=Write-Protected 53=?Illegal Quantity 5=End of Data 69=?Overflow 6=File Not Found 77=?Out of Memory 7=Volume Mismatch 90=?Undef'd Statement 8=I/O Error 107=?Bad Subscript 9=Disk Full 120=?Redim'd Array 10=File Locked 133=?Division by Zero 11=Syntax Error (DOS) 163=?Type Mismatch 12=No Buffers Available 176=?String Too Long 13=File Type Mismatch 191=?Formula Too Complex 14=Program Too Large 224=?Undef'd Function 15=Not Direct Command 254=?Reenter 255=Ctrl-C Interrupt	
224-225 X of Last HPLLOT (0-279)	\$E0.E1
226 Y of Last HPLLOT (0-191)	\$E2
228 HCOLOR Code	\$E4
0=0, 42=1, 85=2, 127=3, 128=4, 170=5, 213=6, 255=7 Test: HGR: FOR J=0 TO 255: POKE 228,J: HPLLOT 0,0: CALL-3082: NEXT	
230 Hi-Res Plotting Page	\$E6
POKE 230,32=Pg.1; POKE 230,64=Pg.2; POKE 230,96=Pg.3	
231 SCALE	\$E7
Note: SCALE=0 is equivalent to a SCALE of 256.	
232-233 Shape Table Start Address \$E8.E9	
234 Hi-Res Collision-Check	\$EA
Example: XDRAW a shape. If PEEK(234) =0 then the shape started at a non-black hi-res point.	
241 SPEED	\$F1
Note: PEEK(241) is 256 minus the current SPEED.	
243 FLASH Mask	\$F3
249 ROT	\$F9

Display Switches

DECIMAL (with neg. equivalent)	HEX
49232 (-16304) Graphics	\$C050
49233 (-16303) Text	\$C051
49234 (-16302) Full-Graphics	\$C052
49235 (-16301) Split-Screen	\$C053
49236 (-16300) Page 1	\$C054
49237 (-16299) Page 2	\$C055
49238 (-16298) Lo-Res	\$C056
49239 (-16297) Hi-Res	\$C057

Note: Activate display switches with a *Poke Location,0*. Example: **POKE 49232,0** switches to Graphics display.

Keyboard, etc.

DECIMAL (with neg. equivalent)	HEX
49152 (-16384) Read Keyboard ...	\$C000
49168 (-16368) Clear Keyboard ...	\$C010

Example: 10 KEY=**PEEK(49152)**: IF KEY<128 THEN 10
20 **POKE 49168, 0**
30 PRINT "KEY: "; CHR\$(KEY-128)

49200 (-16336) Click Speaker	\$C030
--	--------

Example: FOR A=1 TO 99: BUZZ=**PEEK(49200)**: NEXT

49249 (-16287) Button #0	\$C061
---	--------

Paddle-0 Button or Apple IIe Open-Apple key.

49250 (-16286) Button #1	\$C062
---	--------

Paddle-1 Button or Apple IIe Solid-Apple key.

49251 (-16285) Button #2	\$C063
---	--------

Example: If **PEEK(49249+P)** is greater than 127, then Paddle Button #P is pressed (or disconnected).

Useful Pokes

POKE 33,33 Scrunches listings.
POKE 2049,1 Makes 1st line list repeatedly.
POKE 40193,PEEK(40193)-N: CALL 42964 Moves DOS buffers down N*256 bytes.
POKE 44452,N+1: POKE 44605,N Allows N file names before catalog pause.
POKE 44505,234: POKE 44506,234 Exposes deleted file names in catalog.
POKE 44578,234: POKE 44579,234: POKE 44580,234 Cancels returns after file names.
POKE 44596,234: POKE 44597,234: POKE 44598,234 Cancels catalog pause.
POKE 44599,234: POKE 44600,234 Stops catalog at each file name; wait for key.
POKE 49107,234: POKE 49108,234: POKE 49109,234 Prevents language card reload.
IF **PEEK(-637)**=255 then Apple IIe.
IF **PEEK(-637)**=223 then Apple II or II+.
IF **PEEK(-637)**=234 then Franklin Ace or...?

Notes

Apple's memory consists of 65,536 BYTES, numbered 0 to 65535. Every byte has a VALUE in the range 0-255.

- You can **PEEK** or *look at* the value in Byte number-B with the command— **PRINT PEEK(B)**
- You can usually **POKE** a new value-V into Byte-B with the command— **POKE B,V**

Values higher than 255 must be stored in *two bytes*.

- To look at the value in consecutive bytes **B1-B2**— **PRINT PEEK(B1)+PEEK(B2)*256**
- To **POKE** a new value V (0-65535) into bytes **B1-B2**— **POKE B1, V-INT(V/256)*256 and... POKE B2, INT(V/256)**

Note: Program listings will often reveal thousands of Peeks and Pokes not listed on this chart. *Any memory location* may be Peeked or Poked. **POKES** are often used to write *machine-language routines* that are activated with **CALLs**; the possibilities are infinite.



Beagle Bros™
Micro Software Inc.

DOS 3.3

DECIMAL	HEX
976-978 Re-enter-DOS Vector	\$3D0.3D2
1010-1012 Reset Vector	\$3F2.3F4
Example: POKE 1012, 0 makes Reset boot. (POKE 1010,56 to restore normal Reset function.)	
1013-1015 Ampersand Vector	\$3F5.3F7
Examples: POKE 1014, 165: POKE 1015, 214 makes "&" LIST. POKE 1014, 110: POKE 1015, 165 makes "&" CATALOG. POKE 1014, 18: POKE 1015, 217 makes "&" RUN.	
1016-1018 Ctrl-Y Vector	\$3F8.3FA
42350 Catalog-Routine	\$A56E
Example: CALL 42350 catalogs a disk.	
40514 Greeting Program Run-Flag	\$9E42
POKE 40514,52 and INIT a disk. When booted, DOS will attempt to <i>BRUN</i> the greeting program. POKE 40514,20 for <i>EXEC</i> .	
43140-43271 DOS Commands	\$A884.A907
43378-43582 Error Messages	\$A972.AA3E
43616-43617 Last Blood Length	\$AA60.AA61
43634-43635 Last Blood Start	\$AA72.AA73
43698 DOS Command Character	\$AAB2
43607 Maxfiles Value	\$AA57
43697 Maxfiles Default Value	\$AAB1
43624 Drive-Number	\$AA68
Example: POKE 43624, DR changes disk I/O to Drive DR.	
43626 Slot-Number	\$AA6A
Example: POKE 43626, SL changes disk I/O to Slot SL.	
44033 Catalog Track Number	\$AC01
44567 #Chars.-1 in Catalog File Name ...	\$AE17
45991-45998 File-Type Codes	\$B3A7.B3AE
45999-46010 Disk Vol. Heading ...	\$B3AF.B3BA
46017 Disk Volume Number	\$B3C1
46064 Number of Sectors per Track	\$B3F0

To read a disk's Track-**TR**, Sector-**SE** into RAM location 38400—
10 REM DANGER! Back-up valuable data.
11 REM (If you ruin a disk, forget where you read this.)
20 **POKE 768,32: POKE 769,227: POKE 770,3: POKE 771,76: POKE 772,217: POKE 773,3**
30 **POKE 47083,0: POKE 47088,0: POKE 47089,150: POKE 47090,0: POKE 47091,0: POKE 47092,1: REM (Poke 47092,2 writes to disk; could be fatal.)**
40 **POKE 47084,TR: POKE 47085,SE: CALL 768**
50 FOR L=38400 TO 38655: PRINT L-38400, **PEEK(L): NEXT**

Useful Calls

DECIMAL (add 65536 for pos. equivalent)	HEX
CALL-25153 Reconnect DOS	\$9DBF
CALL-3086 Clear hi-res screen to black	\$F3F2
CALL-3082 Clear hi-res to recent hcolor	\$F3F6
Example: HGR: HCOLOR=5: HPLLOT 0,0: CALL-3082	
CALL-2613 Hi-res coordinates to Zero-Page ..	\$F5CB
Example: The X and Y starting coordinates of the <i>next</i> shape table DRAW or XDRAW may be determined with a CALL-2613 and X= PEEK(224)+PEEK(225)*256: Y=PEEK(226) .	
CALL-1438 Pseudo-Reset	\$FA62
CALL-1370 Boot	\$FA87
CALL-1321 Display all registers	\$FAD7
CALL-1184 Print "Apple II" (newer Apples)	\$FB60
CALL-1036 Move cursor right	\$FBF4
CALL-1008 Move cursor left	\$FC10
CALL-998 Move cursor up	\$FC1A
CALL-958 Clear text from cursor to bottom ..	\$FC42
CALL-922 Move cursor down	\$FC66
CALL-868 Clear text-line; cursor to right	\$FC9C
CALL-756 Wait for keypress	\$FDC0
CALL-678 Wait for Return	\$FD5A
CALL-676 Bell; wait for Return	\$FD5C
CALL-657 Input; accepts commas & colons ..	\$FD6F
10 PRINT "NAME (LAST, FIRST) .": : CALL -657 20 A\$="" : FOR X=512 TO 767: IF PEEK(X)<>141 THEN A\$=A\$+CHR\$(PEEK(X)-128): NEXT X	
CALL-468 Memory move	\$FE2C
Note: Here is a Basic memory-move that does not use a <i>direct</i> CALL-468 . Let OS & OE equal the <i>Old</i> -location Start & End and NS & NE equal the <i>New</i> Start & End. Now GOSUB 5000— 5000 N=OS: LOC=60: GOSUB 5020: N=OE: LOC=62: GOSUB 5020: N=NE: LOC=64: GOSUB 5020: N=NS: LOC=66: GOSUB 5020 5010 POKE 768, 160: POKE 769, 0: POKE 770, 76: POKE 771, 44: POKE 772, 254: CALL 768: RETURN 5020 POKE LOC, N-INT(N/256)*256: POKE LOC+1, INT(N/256): RETURN	
CALL-415 Disassembler	\$FE61
Note: Poke start address at locations 58-59 before Call.	
CALL-211 Bell; print "ERR"	\$FF2D
CALL-198 Ring bell	\$FF3A
CALL-155 Bell; enter monitor	\$FF65
CALL-151 Enter monitor	\$FF69
CALL-144 Scan input buffer	\$FF70
100 A\$=""300: A9 C1 20 ED FD 18 69 01 C9 DB D0 F6 60 300G D823G" 110 FOR X=1 TO LEN(A\$): POKE 511+X, ASC(MID\$(A\$,X,1))+128: NEXT 120 POKE 72, 0: CALL -144	

*APPLE® IS A REGISTERED TRADE MARK OF APPLE COMPUTER INC.

Beagle Bros sells useful and entertaining Utilities, Games and Publications for Apple II, II+ and IIe Computers.
To get on a really good mailing list, write or call:
BEAGLE BROS, 4315 Sierra Vista, San Diego, California 92103 / Phone 619-296-6400